THAT WHICH IS CLAIMED IS:

1. A secured microprocessor comprising a rights allocation system for the allocation, to programs executable by the microprocessor, of permanent access rights to certain zones of the memory array of the microprocessor, wherein the rights allocation system comprises means to confer, on a sub-program shared by at least two programs, temporary rights of access to certain memory zones when the sub-program is called by one of the programs, the extent of the temporary rights being a function of the program calling the sub-program.

2. A microprocessor according to claim 1, wherein the rights allocation system comprises means to temporarily confer the rights of the calling program on a sub-program.

3. A microprocessor according to one of the claims 1 and 2, wherein the rights allocation system comprises means to furthermore confer permanent rights on a sub-program that are independent of those of the calling program.

4. A microprocessor according to one of the claims 1 to 3, wherein the rights allocation system comprises:

- a rights allocation table laid out for the reception of an identification code of a program or a sub-program at a first input and an identification code of the memory zones designated by the current addresses flowing in the address bus of the microprocessor at a

second input, and;

- means for the application, to the first input of the rights allocation table, during the execution of a sub-program, of an identification code of the program that has called the sub-program.

5. A microprocessor according to claim 4, wherein the rights allocation system comprises means for the simultaneous application, to the first input of the rights allocation table, during the execution of a sub-program, of an identification code of the sub-program being executed and an identification code of the program having called the sub-program.

6. A microprocessor according to claim 5, wherein bits of the identification code of the sub-program being executed and bits of the identification code of the program having called the sub-program are combined by a logic function before being applied to the first input of the rights allocation table.

7. A microprocessor according to one of the claims 4 to 6, wherein the rights allocation system comprises:

- a first latch for the storage, during the execution of an instruction, of the identification code of the program or sub-program being executed;

- a second latch having its input connected to the output of the first latch, laid out to store the identification code of a program being executed when the microprocessor switches over into a sub-program, to form the identification code of the program that has called this sub-program, the second latch being reset

when the microprocessor exits from the sub-program.

8. A microprocessor according to claim 7, wherein the loading and resetting of the second latch are controlled by an address decoder receiving, at input, the current addresses flowing on the address bus, laid out for the application of a loading signal to the second latch when the address of the first instruction of a sub-program is detected, and to deliver a resetting signal to the second latch when the address of the last instruction of the sub-program is detected.

9. A microprocessor according to one of the claims 4 to 8, wherein the identification codes of the memory zones designated by the current addresses and the identification codes of the programs and sub-programs being executed are delivered by an address decoder receiving, at input, the current addresses flowing on the address bus.

10. A microprocessor according to one of the claims 1 to 8, wherein the rights allocation system sends out a violation signal when an address present at the address bus does not correspond to the rights permanently or temporarily allocated to the program or sub-program being executed.

11. A microprocessor according to claim 10, wherein the address violation signal is processed by an interrupt decoder to send the microprocessor into an address violations processing sub-program.